Augmenting Whiteboard Interaction in the Classroom

Henning Breuer
University of Applied Sciences, Potsdam
Potsdam, Germany
breuer@fh-potsdam.de

Nelson Baloian
Universidad de Chile
Santiago de Chile
nbaloian@dcc.uchile.cl

Abstract: Interactive whiteboards in the classroom are usually used as presentation media with annotating features - combining slide presentation with graphic chalkboard functionality. Pedagogic and situational limitations of this approach are hardly being discussed or even evaluated. Through literature review, observation, and interviews with teachers who use interactive whiteboards in their lectures, we identify current limitations of standard software. We derive requirements for a software environment that exploits the potentials of interactive whiteboards in the face to face classroom setting, integrating various applications within a unique technical and visual framework. We present two prototypes, one of which has been evaluated by students in a Java Programming class. Finally, we propose a gesture-based interaction paradigm and a hierarchical semantic for a “DeepBoard” system currently under development. It allows flexible creation, presentation and documentation of learning materials, supplying a sense of “depth” to the experience with interactive whiteboards.

Many efforts to support learning with computing technology focus on individual training and the design of distance learning systems. Still, most of the learning activities take place in the traditional classroom, thus in a face to face situation. Instead of trying to redesign the traditional educational process and the rich interaction that may take place within it, our approach to integrate technology into the classroom intends to facilitate the ongoing exchange of information and collaborative learning processes.

The use of blackboards as shared projection surfaces is as old as the classroom itself and may live as long. Their primary functions are to guide the lesson through different phases, to focus the views and attention of the audience towards a central stage, to document the lesson (enabling the students to rewrite its content for later studies), and to increase learning and understanding by providing structure (Leisen & Mentges 00). They may help to visualize spatial relations in terms of logical images, trying to connect distant concepts. The first electronic “LiveBoards” were developed by Xerox PARC (Stefik et al. 87). Many hardware vendors followed. Basically they offer a secondary projection of graphical annotation onto a prefabricated presentation. Every user interacts with these two formats or layers, which do not interact or know about one another. We try to proceed or overcome the conceptual gap arising from this division. This paper reports on an ongoing series of research activities. We start by analyzing current usage of interactive whiteboards in the classroom setting. From the analysis we come up with guidelines and requirements for enhanced interaction. These should allow for being able to easily create and modify learning content during the time in class. Applying these requirements, software for interactive whiteboards based on the JavaFreestyler environment has been developed. This software was tried out in a programming class, and evaluated with a questionnaire filled out by the students. In the last part we present a system called DeepBoard. It fully supports these guidelines by providing users with a gesture based paradigm to create and manipulate learning materials in class. DeepBoard may be used for teaching in a face to face setting, but also to enable students to contribute content in distant learning scenarios.

Related research on interaction with large-screen interfaces and interactive whiteboards often focused on supporting collaborative workgroups and business meetings. Most “presentation software” usually being used in class like Microsoft PowerPoint, or Smart Notebook, was not specifically designed for educational purposes (Llanza 04). Few educational lecture tools are able to mix drawing and discussing, and the ability to generate high quality documents or graphics easily, without interrupting the class. One of these is FreeStyler (Hoppe & Gassner 02): It combines “Concept Mapping” tools with archiving and retrieval functions. These allow the building and accessing of a group- or “Corporate Memory”.

Beside the retrieval aspects, the system supports structuring and representing different kinds of knowledge. A palette of different object types and relationships (as well as annotations and free handwriting) provides a visual language in the form of semantic networks. FreeStyler has been frequently used in this context, but it has not been developed with the primary goal of presenting and developing learning material in classroom settings. It has not yet been tested in this context formally.

Within the business domain, cooperation in multimedia-offices and CSCW-environments has been studied. Productivity tools for primarily informal workgroup meetings (e.g. Ishii et al. 93) or gesture based support for object-oriented modeling (Damm et al. 00) have been developed. An early application for the Xerox LiveBoard called Tivoli (Moran et al. 97) explored the transition from freeform writing to structured interaction. It allowed groups of users collaborating in real time to flexibly organize and arrange materials on the board through direct manipulation of boundaries and recognition of “implicit structures” by the system. Some works examine gesture analysis in pen-based interfaces. Architecture named “Flatland” has been developed. It uses freeform strokes as basic input and output primitives for office whiteboards, flexible screen segmentation and pluggable applications for different segments (Igarashi et al. 00). Close lines of research deal with display walls for information visualization in control centers, shared displays in meeting rooms, and large screen metaphors.

In the educational field, opportunities for free drawing in Kindergarten have been explored through observation (Ovasaka et al. 03). Initiatives for collaborative learning in school (Levy 02, Virtual Learning 02) and the computer-integrated classroom (Baloian et al. 02) have been reported. Focusing on reuse of teaching materials, the eClass (or Classroom 2000) project introduced the notion of “teaching and learning as multimedia authoring” (Brotherton & Abowd 04). In trying to optimize the use of e-boards in classrooms, different setups of the environment and input devices like tablet-PCs, PDAs and Laptops have been considered. Hardly anyone questioned the quality of the board-show itself, taking the windows oriented desktop projection and annotation philosophy for granted.

Instead of setting up physical environments, we looked at the use and potentials of e-boards in classroom settings to question, define and evaluate new designs for interactive whiteboard interfaces. A central idea was to reduce the teacher’s performance to his interaction with the students, off- or on-board. With silent support of the board the teacher designs and moderates the ongoing communication in class. He may present or extract content from preexisting media like internet or disks, create new parts from scratch, filter and collect student discussions, and relate the parts by manual representation on the whiteboard. In other parts of the course he may grant access to the board to the students as a group or individually. Students and teachers interact to co-construct the course of their education.

Current Limitations of Whiteboard Use and Guidelines for Whiteboard Interaction

In order to identify the main limits of the current use of interactive whiteboards in class we started with a literature review and extensive observation of in-class activities. Continuously collecting notes, clustering them into emerging categories and reviewing the categories with respect to new observations in programming and foreign language classes, we defined preliminary categories for software-related limits. Additionally semi-structured interviews with teachers who used interactive whiteboards in their classes were being conducted and analyzed. These interviews addressed preparation, conduction and follow-up work of classes. In all of these classes standard software like Microsoft PowerPoint, Smart Notebook and standard web browsers was used to present and annotate mainly prefabricated materials. The following limits of this approach have been identified:

Large screen requirements: The desktop metaphor does not suit for body size touch interfaces. Since touch interfaces do not provide the full control of a mouse and keyboard, the resulting interaction becomes awkward. Even with these additions, conventional GUI desktop interfaces are not well suited for use with a large vertical screen. Taking advantage of the possibilities provided by resolution and physical size of the display, arm length interaction designs are required (Guimbretiere et al. 01). A wide spectrum of information types must be incorporated into a coherent visual environment. Various parts of potentially large amounts of interrelated information (such as in maps) may have to be brought to the center or the peripheries of the audiences attention.

Visual presentation on the board: In contrast to desktop interfaces addressing individual users, interactive whiteboards address two distinct groups of users: presenters and their audience. Taking into account size and different distances of the board as well as the vivid context of the class, the different foci of attention for presenters and audience and their
unique perspectives have to be considered. Hoppe et al. (93) refer to the limited quantity of legible information that can be displayed on one surface, as well as to the problem of excessive “windowing” and switching from one to another. Mynatt et al. (00) discovered that people create and maintain multiple clusters of contents on their whiteboard, typically an average of five segments for space distribution. A common complaint from most whiteboard users was the continual challenge to find usable space amongst content that they didn't want to erase. A “disorientation problem” (Scaramalia et al. 94) comes up when too much information is being displayed that is relevant to the presenter, but not to the audience. Finally it has to be considered that the teacher may want to prepare his class on his desktop, and then has to anticipate the functioning of his prepared materials within the enlarged visualization on the board in the vivid context of the class.

Teacher’s performance and software environment: Spending too much time in activities, which are not directly related to teaching, like typing long commands or queries, or searching for files, may break the dynamic flow of the lecture and distract the attention of the audience. A frequent mistake of teachers is to turn their backs to the audience while working with chalkboards and overhead projection. We observed some teachers running back and forth between the e-board, keyboard and mouse. Not only their emotive-evocative expression and attention, but also the students’ attention and the interaction between students and teachers were disrupted by distracting technology. This could be avoided by enabling all interaction on the interactive whiteboard as the unique input and output device. Different kinds of applications, e.g. to show sketches or renderings, perform live applications and simulations, or to collect and manipulate student input should be integrated within a consistent software environment that does not absorb the students attention.

Constructivist interaction in class: Whiteboards have been criticized for promoting a teacher-centered approach giving a passive role to the learners (Levy 02, Virtual Learning 02). Constructivist approaches to teaching and learning propose open, but problem-oriented classroom activities. Support of self-directed and cooperative student activities moderated by the teacher is being considered favorable for relevant learning experiences (Mandl & Reimann-Rothmeier 98). Instead of presenting pre-fabricated learning content and transferring his knowledge to the students the teacher acquires the role of a moderator of his students who co-construct their knowledge with his support, using various media. The presentation- and-annotation philosophy of standard interactive whiteboard software does not support these kind of flexible activities well. Instead it suggests operationally proceeding slide by slide through a pre-fabricated presentation. Co-creation and reuse of content and visualization of sociological images as they emerge are limited. The graphical implementation of the interaction between students and teachers were disrupted by distracting technology. This could be avoided by enabling all interaction on the interactive whiteboard as the unique input and output device. Different kinds of applications, e.g. to show sketches or renderings, perform live applications and simulations, or to collect and manipulate student input should be integrated within a consistent software environment that does not absorb the students attention.

Remote lectures: Some campuses are provided with a complete set of tools to support remote lectures, from videoconferencing systems to interactive whiteboards. Even though the interaction with the whiteboard may be projected into the local classroom with the teacher and into the remote classroom, students in the remote classroom do usually not have access to an interactive whiteboard. Their participation is therefore usually limited to asking questions to the teacher through the videoconferencing system. Having constructivist approaches to learning and teaching in mind, access to the interactive whiteboard in remote classrooms would empower the student’s self-directed learning activities by contributing via their laptops or writing on the interactive whiteboard in the remote location.

Notebook access: Students increasingly carry their computer notebooks into the classroom. Currently class presentation systems are rarely capable of interacting actively with students’ notebooks allowing automatically downloading materials, or repeating teacher actions. Once the students gain access to the system, they should be able not only to receive and read materials, but also to interact with the rest of the class. For example, a teacher could grant the main screen control to a student, local or remote, allowing him to solve some exercises, explain some concepts, etc., on the interactive whiteboard or accessing with his notebook. When sufficiently structured didactical concepts are applied, this may enrich teacher-student and student-student communication, even in remote conditions.

Taking these constraints into consideration we derived the following design guidelines to exploit the capabilities of interactive whiteboards in the classroom setting.

• To avoid changes of contexts the system should offer an integrated framework for all actions on the board, such as presenting and developing learning materials, or documenting in-class activities.

• The system should provide a high degree of flexibility in adding, changing, storing, and retrieving documents and relations.

• Supplementary tools for accessing whiteboard content from remote locations and student notebooks should be provided.
An Interface for JAVA Programming Classes and its Evaluation

As a starting point for our project we selected a Java programming lecture to solve some of the problems outlined above and to respond to the first two guidelines. As a special case of these problems programming classes involve the demonstration and execution of code. To do so with standard whiteboard software the teacher currently needs to handle four frameworks: a file manager, a command prompt, a text editor and a PowerPoint presentation in separate windows, switching back and forth between them. To create an integrated framework we brought together all four frameworks within one application. It was implemented using the Java FreeStyler software developed to manage documents within the computer-integrated classroom (Baloian et al. 02).

FreeStyler is a tool for creating “active documents” with interactive elements. Elements can be chosen from a palette and combined with one another in different kinds of relationships, for example, to associate the output of a certain element to the input of another. New palettes containing certain domain-specific interactive elements (nodes) and relationships among them (arrows) are defined by extending the basic types of node and link provided by the system. FreeStyler also permits free-hand annotation over the whole workspace where nodes and links are combined, which makes this tool suitable for presenting active documents on an electronic board.

For our Java lecture we defined a new “Java-palette” which includes nodes and links designed to present, comment, compile and execute Java programs in the same page of an active document without changing context (see Fig. 1). The Code Node contains a text editing area to show and manipulate java programs. It allows compiling code, executing Java programs, importing of code files from disk, and specifying program parameters within one framework. A status bar indicates the process of compilation. In addition, class nodes allow including existing class or jar files for use in the compilation and execution of the code. Finally, a Rich text node allows including formatted textual presentation in the document along with the code (diverse pen styles with color, highlight, resize). With these nodes the presentation can be flexibly modified and extended within the classroom situation without changing the framework. Programs can be easily changed, compiled, and demonstrated. New examples may be developed. Free hand writing can be drawn over workspace and nodes. The environment may be easily accustomed to all classes which afford simulation or calculation with measures.

![Figure 1: Java FreeStyler presentation interface](image)

To evaluate this design we set up three scenarios. All scenarios were used in equal parts during a one week intensive Java Lecture at the Graduate School of Global Information and Telecommunication Studies, Waseda University, Tokyo in January 2004.

- The media for the first scenario included MS PowerPoint, as a slides-based presentation tool, Explorer, as a file manager, Command Prompt MS-DOS to enter the necessary commands, and Notepad as a text editor.
- The second scenario applied the same technologies plus an interactive whiteboard.
The third scenario was reduced to the use of an interactive whiteboard as interface device and Java FreeStyler.

After the end of the class the 21 mainly novice participating students answered a questionnaire on general information, class conditions, exposition quality in the different environments, and general evaluations. About 95 percent of them considered the interactive whiteboard as a useful support in class. For each of the scenarios the students rated on a 5-point scale their level of agreement with 19 statements. The scale ranged from -2 to +2 with negative values indicating rejection and positive values indicating agreement with the respective statement. To avoid a response set in the pattern of users, answers to some statements were positively phrased and others negatively phrased. For the analysis all were recoded to indicate positive assessments.

Since four items served explorative purposes for future investigations they were left out within the following analysis. The other items were summed up in six categories which (except for “preparation”) proved to be sufficiently reliable. “Preparation” refers to statements regarding possible disturbances caused by setting up and calibrating the system before the class. “Presentation” comprises items relating to the visual presentation of content, such as adequate colours, fonts, and layout. Items yielding to the flexibility provided by the board for the ongoing flow of the class, i.e. the flexibility to create, modify, or highlight content, and to allow situational adaptation to unexpected questions, were summed up in “flexibility”. The category “highlighting” referred to highlighting features to focus attention. Absence of disruption sums up statements on possible disruptions through the necessary use of extra devices like keyboard and mouse. The last category referred to the usefulness of documents created in class for “follow-up” work by the students.

In order to test conservatively, nonparametric analysis was applied. The nonparametric Friedman test for dependent samples revealed significant group differences for the scales presentation, flexibility, highlighting and follow-up. To check which of the scenarios differ significantly from one another, pair-wise comparisons were conducted afterwards. Bonferroni adjustment was applied and the alternative hypothesis was accepted for $p<0.016$ (equals 0.05 divided by three). Analyses showed that scenario three was evaluated more positively than the others regarding flexibility. Regarding highlighting both whiteboard scenarios were perceived similarly, but significantly better than the first scenario without interactive whiteboard.

Since four items served explorative purposes for future investigations they were left out within the following analysis. The other items were summed up in six categories which (except for “preparation”) proved to be sufficiently reliable. “Preparation” refers to statements regarding possible disturbances caused by setting up and calibrating the system before the class. “Presentation” comprises items relating to the visual presentation of content, such as adequate colours, fonts, and layout. Items yielding to the flexibility provided by the board for the ongoing flow of the class, i.e. the flexibility to create, modify, or highlight content, and to allow situational adaptation to unexpected questions, were summed up in “flexibility”. The category “highlighting” referred to highlighting features to focus attention. Absence of disruption sums up statements on possible disruptions through the necessary use of extra devices like keyboard and mouse. The last category referred to the usefulness of documents created in class for “follow-up” work by the students.

<table>
<thead>
<tr>
<th>Category</th>
<th>S 1</th>
<th>S 2</th>
<th>S 3</th>
<th>Chi-Square</th>
<th>Significance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Preparation</td>
<td>2.21</td>
<td>1.98</td>
<td>1.81</td>
<td>2.655</td>
<td>0.265</td>
</tr>
<tr>
<td>Presentation</td>
<td>1.67a</td>
<td>2.00a</td>
<td>2.33a</td>
<td>6.644</td>
<td>0.036</td>
</tr>
<tr>
<td>Flexibility</td>
<td>1.67a</td>
<td>1.74a</td>
<td>2.60b</td>
<td>14.952</td>
<td>0.001</td>
</tr>
<tr>
<td>Highlighting</td>
<td>1.48a</td>
<td>2.10b</td>
<td>2.43b</td>
<td>13.079</td>
<td>0.001</td>
</tr>
<tr>
<td>Absence of Disruption</td>
<td>1.90</td>
<td>1.86</td>
<td>2.24</td>
<td>2.868</td>
<td>0.238</td>
</tr>
<tr>
<td>Follow-up</td>
<td>1.71a</td>
<td>2.02a</td>
<td>2.26a</td>
<td>9.172</td>
<td>0.010</td>
</tr>
</tbody>
</table>

Table 1. Mean rank values for the scenarios S1, S2, and S3, chi-square and significance. Differing indices (a versus b) represent significant differences.

Since highlighting some of the contents being presented in class is a typical use of the interactive whiteboard, this last result may depend on the teacher applying this feature frequently when working with the interactive whiteboard, independent of the software being used. Students appreciated this feature as a useful means to focus attention. With respect to the initial observations of in-class activities it was surprising not to have found significant results with respect to disruption of the normal flow of the class through extra devices. In respect to these disruptions, but also to the other categories that did not show significant results, extra development efforts are needed to demonstrate more obvious advantages of our solution. A promising result for the evaluation of these works is the high flexibility that participants attributed to the system applying the FreeStyler Environment, compared to the standard software. It indicates that the system may in fact seamlessly guide the normal flow of the class by providing flexible support for the creation and modification of learning materials, and to flexibly handle unexpected occurrences. For further studies it will be desirable to include more participants working for a longer time interval with one of the different scenarios. However, this first test supports the hypothesis of the potentials in flexibility in our design approach.
Hierarchical Writing on the DeepBoard and Gesture-Based Interaction

Trying to further close the conceptual gap between presentation and annotation we are developing a relational writing system called DeepBoard. This system allows flexibly creating, modifying and managing content and avoiding changes of context by implementing a three dimensional semantic graph. The concept of hierarchical writing provides semantic relation between parts of the contents drawn on the e-board’s screen. This also differentiates this approach from the Flatland interface mentioned in Igarashi et al. (00). Hierarchical writing with the implemented system allows defining node objects on the screen by drawing the height and width of a rectangle in a single gesture. The system draws a complete rectangle and defines a new node containing all the objects drawn before inside the area of the rectangle. It is always possible to add new content to the node. Objects are defined as a collection of strokes. The strokes are described as a collection points united by a straight line. After each drawing, the recognition system will translate the input into graphical objects based on vector graphics, instead of capturing the whole screen bitmap. Drawing a stroke which goes from one node to another connects them internally defining a father-son relationship between the nodes.

Gestures on the screen are automatically detected as such and interpreted semantically by the system. For easy retrieval and follow-up work reusable materials are stored with the semantic structure. Seen from the users’ perspective, the final goal would be to reduce all necessary interaction to the moderating gestures and documentary writing on the screen. The hierarchical structure in Fig. 2 shows the result of free-hand writing, importing text, and relating the objects by gesture interaction on the screen. By this interaction the system allows the users to give a semantic, hierarchical structure to the content that is normally free-hand written and stored as a picture. Each element (node) may contain recursively another structure with connected objects.

As stated in Moran et al. (97) there are inherent problems with gestures, as they are abstract and thus their meanings are arbitrarily defined. There are many “gestures” which are already more or less common for mouse-based or pen-based interfaces. For example, a click (tap on the whiteboard surface) is normally used to select an element. A double click may be used to “open”. As these gestures are easy to reproduce conventions we are going to keep the meanings for these actions in the whiteboard scenario. However, new ones should be defined like those for creating new elements or deleting them, since normally these are done by selecting options in a context pop-up menu and these kinds of gestures are not easy to produce with the e-board.

In DeepScreen there are two modes to operate the whiteboard’s interface: command mode and write mode. Users can switch from one mode to another mode by pointing the electronic pen to the whiteboard twice rapidly in an area where no object is located. When the status of the whiteboard is in command mode users may perform the following actions:

- Select an area and define a node with its content by drawing the height and width of a rectangle in a single gesture. The system draws a complete rectangle and defines a new node containing all the objects within the rectangle
- Select a node by tapping once on the node.
- Delete a node by selecting a node or an area and marking it with a cross.

![Figure 2: Hierarchical Writing on the DeepBoard](image)
• Link a node with another node, forming a “parent-son” relationship: This is done by drawing a line between two nodes. The starting node is defined as the son and the ending node is the parent. The stroke is replaced by the system by a straight arrow which will always connect the two nodes, even if they are moved. As more objects are connected, these relations are interpreted as a hierarchical structure.

• Unlink two nodes: Doing the same gesture between nodes which have already a link defined between them.

• Move a node by selecting it with a single tap and then dragging it.

• Opening a node: The content of any of the objects can be seen and edited as a whole screen by double-clicking over it. The node will then take the space of the whole screen and more content can be added. Within an open node new nodes may be created. This operation can be done recursively.

• Closing a node: By double clicking again the node will “close” and the whole structure will be shown again. Only the left-upper area of the whole node will be the visible. This corresponds to the original size of the node when it was defined.

More complex gestures include zooming, creating and managing lists, tables, and graphs. The gesture recognizing function can be turned off for a while. In this mode users may edit (write, cut/paste, delete, undo/redo) content on the whiteboard. They are provided a palette to change the color of the context and change the size of the pen. Text and/or pictures form existing files can be added to a node as content. The nodes can be automatically rearranged over the screen by selecting a “root” object and clicking over an icon in the palette. This will rearrange the location of the nodes and shape of the links in order to display them as a symmetric hierarchical structure.

Starting from literature review, observation of classroom activities in programming and foreign language classes, and semi-structured interviews with teachers, we identified typical problems with the current use of interactive whiteboards in education. We derived guidelines for interaction and implemented a Java FreeStyler environment. Evaluation of the system showed significantly higher appreciation of the system’s potential to flexibly adapt to the dynamics of the class depending on situational changes. In the next step we developed a system called DeepBoard that realizes a gesture-based paradigm and provides for remote access.

For creating such environments it is not always necessary to create a new system. PowerPoint 2003, for example, may include different programs like UltraEdit or Eclipse in the presentation. However, these functionalities are neither tailored to (specific) lectures, nor do they allow storing the whole (programs, presentation and annotations) as a single reusable document for follow-up work. Remote and laptop access are also not easily available.

Instead of promoting a single piece of software to fit all possible usage scenarios for interactive whiteboards in different settings, we sympathize with tailoring the interaction design of the whiteboard to the activities in a specific classroom setting, or even specific kinds of didactics and lessons, like programming or language classes. For different situations such as a fair show, an exhibition, or use in a museum, quite different software support might be needed.

More than proposing this particular FreeStyler instantiation of the software or the DeepBoard system, our idea is to exemplify and evaluate an example of a central design principle: To enable interaction in an integrated whiteboard environment, avoiding changes of context, and comply with the requirements of present and remote problem-oriented learning. Following this principle on to the DeepBoard system, we aim at going back to classical chalkboards, “purifying” or economizing the potential of e-boards in class by reducing all interaction to a single point of contact, and transforming e-boards from a writing surface into a seamless interaction space within the classroom setting. Without prescribing didactical concepts or goals, we aim at enabling flexible, body sized, gesture based interaction in order to support problem-oriented learning by trying to translate requirements of the classroom setting and its social learning activities. Within the envisioned environment, actions on the board orchestrate the course of the class. The whiteboard fully supports the functions of the traditional blackboard and supplies them with semantic depth.

Acknowledgements

We thank Felipe Baytelman, Leandro Llanza and Stephan Buschmann for their programming efforts, Wolfram Luther, Maike Breuer and Paz Aburto for their kind support and feedback on our project and this article.
References


